# To SAN or not to SAN?

## The case for DAS and application servers

By Dick Blaschke and Samek Mokryn

**Introduction**

Storage Area Networks ("SAN") are based on the assumption that STORAGE is separate from COMPUTING. The immediate corollary of this assumption was the introduction of SAN, network cloud that interconnects the servers and storage. However, the introduction of this network brought with it all the problems associated with any network: authentication (who is who), access control (who has access rights), sharing (how to access coherent data in an orderly fashion when data is being modified simultaneously by somebody else), predictable bandwidth (since resources can be seized by somebody else exactly when you need them), partitioning (how to assure that problems in some part of the network do not affect data in adjacent systems), reliability (the larger the system, the more things can, and will go wrong) and security (how to defend the precious data from malicious access). Along with all of this, you have to know what data resides where, since in SAN you can reassign STORAGE parts dynamically. The network topology is dynamic to some degree too. Finally, the network has to support heterogeneous STORAGE and COMPUTING supplied by various vendors. All of these problems, and more, are tackled by the NETWORK MANAGEMENT, in and of itself a powerful and complex entity which is difficult to operate and maintain.

But is all of this necessary?

There is another network, the IP network, which interconnects servers. For the last 30 years the IP network has addressed the same problems, is well established and has a rich assortment of tools and procedures to deal with network operations and maintenance. This article addresses the strengths and weaknesses of SAN, and tries to shed some light on the problems and compromises involved with separation of STORAGE from COMPUTING.

**The short history of SAN**

The story of SAN can be traced to the beginning of the 1960s, when IBM introduced the 360 line of mainframes and the concepts of channel, I/O interfaces and control units. This architecture effectively separated the I/O operations from the main computing activities. Since then, the traditional mainframe installation has been comprised of CPUs (servers), I/O devices represented by their Control Units, and networks of standard I/O interfaces which connect all of these pieces together (Bus & Tag, ESCON, FICON). It was quite natural for "open systems" to adopt this architecture in the 1990s, when non-mainframe computers started to replace the traditional role of mainframes. Bus & Tag was replaced by SCSI, and ESCON was replaced by Fibre Channel, but the general level of abstraction remains the same: devices, volumes, records.

However, contributors to the Fibre Channel standard - the main technology behind the SAN concept - were far more ambitious than their IBM predecessors. They intended to define SAN as an autonomous fiber-optic network that carries all kinds of traffic,

including IP traffic. This approach brought with it all of the aforementioned problems and created direct competition between SAN and IP based LANs and WANs.

**IP based networks versus SAN**

IP based networks encompass LANs (local area networks, like Ethernet and wireless networks), WANs (Wide area networks in various incarnations) and NAS (network attached storage – network file servers). The IP networks were designed from the beginning to facilitate communication between servers or applications. Hence, the level of abstraction of data is related to the applications themselves. This data can be referred to as files, directories, database SQL queries, transactions, email, etc.

The main characteristic of these networks assumes peer–to–peer relationships between endnodes, with the endnodes tasked with the ultimate responsibility of assuring the content and validity of data. Generally, the messages are relatively short and are sent in the form of datagrams, i.e. messages that the network is not responsible for.

In contrast, SANs run almost exclusively SCSI traffic (the only noticeable, but not really different exclusion is FICON mainframe I/O traffic). This traffic is characterized by the master–slave relationship between endnodes, where the servers are the masters, while devices are the slaves. The level of abstraction is relatively low: devices, volumes and data records.

The amount of traffic can be very high and deterministic network behavior is mandatory to assure proper operations.

**Directly Attached Storage (DAS) as an alternative to SAN**

Directly Attached Storage is a group of storage devices directly attached to a single server, or limited group of servers, by some form of SCSI (Fibre Channel or SAS – Serial Attached SCSI). It can be implemented as a very limited SAN. The main difference between DAS and SAN is who is responsible for controlling the network. In the case of DAS, this responsibility lies almost exclusively with the server-based operating system or some dedicated extensions to the operating system. In other words, DAS is never autonomous.

DAS's topology is much simpler, cheaper, and more predictable than large SAN.

DAS is more secure architecturally. The fact that the storage is connected only to the limited computing environment, which can be strictly controlled from the application down to well identified drives and links, makes the security solutions by far simpler. External systems have no way of physical access to data without querying the host systems, while the query system is by definition protected.

Both SAN and DAS can be created with the same degree of reliability. The question is if computing systems utilizing DAS as a connection to storage are more or less reliable than computing systems utilizing SAN as the primary storage architecture. We don't have any research data on this subject, but the logic dictates that DAS should be more reliable. Do you remember Einstein's words: "Make everything as simple as possible, but not simpler"? Since SAN contains additional logic and control elements that are non-existent in DAS and these elements have a limited reliability, their mere presence reduces overall

system reliability. Please note that both SAN and DAS can employ a multi-path configuration, in order to provide an access to data in the presence of errors.

Well structured DAS should outperform an equivalent SAN for vast majority of traffic patterns. Again, simplicity is the main factor. Repetitive tests showed that for many common traffic patterns simple JBODs outperformed most sophisticated storage systems often by as much as 40%.
There is however one more element that is important as well: the predictability of the performance. It is of little interest how fast are system that are underutilized. It is much more important how systems perform under high utilization condition, which is never good. Now, in SAN the traffic is generated by many unrelated servers, which easily can generate local high utilization condition, negatively affecting all participating servers The attempts to prioritize the traffic in SAN are very sketchy at best, and difficult to implement for wide range of traffic patterns, hence the only way to build systems that require very high degree of predictability of performance is with DAS or DAS's equivalent.

Intuitively, DAS has to be much easier to manage reducing the TCO (Total Cost of Ownership). But by how much? If 10%, then no big deal. 50% gets your attention. We have no data on this subject, neither it is easy to measure. There are many factors involved and the weights can be different from installation to installation. Hence the only one thing left to us is the common sense (not exactly very accurate, but taking a different point of view can at least stir the debate on the subject).

The main argument against DAS was that it is too simplistic, that it takes to many server cycles to perform extended functions like RAID, stream replications and snapshots. However with the advances in the technology, very powerful, intelligent controllers can be employed in the servers as coprocessors, or I/O processors, effectively offloading the servers from these tasks. Disks become smarter too, their processors more powerful and caches larger. Today 100 disk system features 8 GBytes of a distributed, record level cache and in near future this number can easily double. Since high-end disks are multi-ported, access to data from various directions is supported, allowing systems to operate in the presence of system errors. This trend allows deploying new advanced storage architectures, which are still DAS in nature but have many expanded functionality features what are today associated with SAN.

In DAS environment, STORAGE is really an integral part of the server complex; hence it can be called storage.

**Why SAN?**
The most often cited reasons to create SAN include:

1. Data sharing
2. Capacity on demand
3. Backup and Restore
4. Organization of IT departments

5. IP network is not a suitable alternative
6. Marketing

Let's try to analyze the points above:

Data sharing

Data sharing through STORAGE was extensively used in the mainframe environment by various applications running on the same CPU complex. Then and now, all resources are very strictly controlled, while the system provides various facilities to assure orderly sharing. For example, I/O operations are executed in chains, or in other words, when one application is accessing storage, the storage is busy to other accesses.

Please note, that in storage-based data sharing, applications cannot hide data in internal buffers, as such data is invisible to others. However, at the time when most mainframe applications were written, memory was at a high premium, hence working directly with storage was an acceptable compromise.

With advances in semiconductor technology, memory has become very large, and given the huge performance penalty of accessing the disk versus accessing the memory, this type of data sharing is generally not used.

Closely related argument used in justifying the SAN implementation is: "SANs enable storage consolidation and reduce the incidence of the same data stored in different locations versus DAS islands. If DAS requires the same data to be stored in multiple places (esp. for non-mainframe systems), the storage costs for DAS are higher."

This is exactly the unfulfilled promise of SAN: "Lets reduce the cost by sharing the data thru SAN". But sharing the disc records is meaningless, if you don't know how data is represented. In other words, without distributed, SAN based file system, with its own directories and other related data structures you cannot share data. There were numerous attempts to define such system, none of each achieving the degree of standardization and recognition required for wide deployment. As a result the data sharing thru storage is practically dead.

Capacity on demand

Even a few years ago, the pace of data growth exceeded the growth of disk capacity and people added disks and replaced older disks at a very high rate. This was a real operational problem. Hence, creating a common pool of STORAGE and providing capacity on demand was a rather novel idea.

However, disk drive capacities (and densities) are growing at about 30-40% annually now - down from ~60% annually throughout the 1990-2003 period. Disk drive performance improves at less then 10% per year. This is the classic access density problem. As a result, levels of drive allocation are falling, around 40% on Unix and Windows.

It is still easier to achieve higher level of storage utilization on SAN than on DAS, since this is a general rule of larger systems versus set of smaller ones. But given the tremendous decrease in the cost of capacity, the importance of high disk utilization is getting smaller, while the reliability, predictability, performance and control are getting

more and more weight. These are exactly the driving forces behind the need of re-examination of the architecture of computer systems in general and storage in particular.

One of the problems limiting system performance is “head starvation”. In other words, there is so much data sitting behind a single read/write head, that there is fierce competition for access to this head. The only one practical solution to increase performance is to have multiple disks, which leads to huge reserved capacity.

Other problem which affects the system’s performance to high degree is the match between data distribution and traffic patterns. This match can be much better controlled in the limited scope of DAS, than in the general, autonomous SAN environment.

Backup and Restore

Well, this is the real killer. The amount of data is vast and growing, while data loss can be catastrophic. Backup must occur often, ideally continuously, while minimally affecting ongoing operations. Hence, designating the backup functions to separate SAN-based functions sounds attractive. But is it?

The obvious reason for making a backup is to have the capability to restore data whenever necessary. However, the data restore must be consistent, i.e. reflect the data in a consistent, not transitory, state. The problem is that only the applications know when the data is in a consistent state, while neither storage nor the operating system posses this information. According to some reviews, some 80% of backups taken will result in an inconsistent restore, resulting in data loss.

Since SAN deals only with records, which is a level of abstraction the application knows nothing about, various proprietary communication schemes are being proposed. However, in the heterogeneous large SAN, with various applications running simultaneously, coordination of all of these various schemes becomes mission impossible.

The disparity in the level of abstraction between how storage perceives the data, versus how applications see it, causes one more serious problem. Since SAN knows only records, but has no information about their content, it cannot judge what data is active and relevant, versus what records are unused or deleted. As an example, whole volume copies which are common in SAN, result in a huge waste in resource utilization and performance.

Organization of IT departments

In most organizations, the infrastructure departments are separate from the application groups. There is even a notion that this separation is absolute and complete, and that the computing infrastructure can be outsourced to other companies. This trend is obviously supported by companies like IBM or EDS.

Further, separate departments tend to specialize in various parts of systems: servers, storage, networks, etc.

Obviously the separation of STORAGE fits this organizational model. While this may sound simple and efficient, is it really?

As of result of this separation, you find different groups speaking different languages, on different levels of abstraction. These groups try to control different aspects of

installations and provide various interface procedures and protocols with the rest of the organization. Given the sheer number of suppliers, incompatible models (even from a single supplier), proprietary interfaces and capabilities, you start to get the picture.
Take for example the backup and restore application discussed above. On the one hand, users wish to have an almost transparent backup and recovery process, so they place this responsibility on STORAGE personnel. But, STORAGE personnel do not have the slightest idea as to what applications are currently running, and what is the active state of each application. As stated previously, this level of abstraction is not available to them, and they are not particularly interested in it. At the end of the day, they are STORAGE personnel, they are interested in storage and not in applications. Their job is to work hard, take snapshots, and make backups. In some controlled environments this separation may even work, but in most cases data restore occurs on a wing and a prayer.
The net result is that serious applications take upon themselves backup and restore responsibilities, which moves the control to the application domain, rather than STORAGE.

IP network is not a suitable alternative

While proponents of ISCSI believe otherwise, it is probably true that IP networks are not a suitable alternative to move storage traffic. However they are ideally suited to move information between servers in the form of application queries.

The golden rule one:
*The higher the level of abstraction, the smaller are bandwidth requirements.*
It is far more efficient to communicate in the form of specific queries, Google searches or files, than in terms of disk records.
The golden rule two:
*In order to move large amounts of data rapidly, simplify your structure.*
As far as bandwidth goes, DAS is a far simpler and more predictable alternative to SAN.

Marketing

There is probably no other subject in all of the history of STORAGE that got as much coverage as SAN. This was due to the notion that STORAGE can be separated from the rest of the computing environment. This movement continues to be supported by all storage system manufacturers. Even in integrated computer companies, like IBM, HP and SUN, the storage divisions are practically separated from the rest of organization. Hence, they need standard, flexible and universal connectivity solutions in order to sell their products.
The problem they all have is the low level of abstraction they are dealing with, which seriously limits the amount of data manipulation they can perform. This limitation and its consequences are being suppressed to this day. However, when you are dealing with vast amounts of data, and the responsibility of proper computing is so great, the need for intelligent, selective data manipulation is critical. This demand requires tighter cooperation between applications and all underlying infrastructure, including STORAGE.

**If not SAN, then what?**
The appropriate answer (in many cases) to this question is a complex of Integrated Application Servers interconnected by IP networks. The simplest and most popular example of such an application server is a file server.

The Application Server is comprised of a set of interoperable applications running on top of an operating system that controls the complete computing infrastructure, including STORAGE.
If this sounds familiar, it should. This was the way large systems were built and till today mainframes are still based on this architecture. There were good reasons for building systems this way.
The proponents of separate STORAGE will be quick to pinpoint the fact that modern storage systems are far more intelligent than their predecessors. While this is true, so long as the level of abstraction remains limited to a storage record, there will be limitations to what you can do with all available processing capabilities.

In interconnected application server architecture, the need for separated I/O network, in addition to the front-end IP network, starts to be questionable. SAN can still be used, but only as a way to statically partition the storage resources, but not as a completely autonomous entity.
The most appropriate storage connection for application servers is DAS, or some limited form of SAN which effectively performs the same function. The most important thing is that the control goes back to the operating systems and applications.

**Final Thoughts**
Current studies show that 70-75% of all SANs are connected to homogeneous (ex: all UNIX, or all Windows) servers. According to Fred Moore: “My belief in the 1990's that true data sharing (not device sharing) between unlike operating systems was going to be huge never materialized. Most SANs are stove-piped as a result (sounds like DAS!!)”

Yes, it sounds as DAS, since sharing the controls and data structures inside homogenous environment is by far easier than in the heterogeneous system. At the end the size of SAN is practically limited to self-controlled system complex, creating effectively DAS. In this article we are going a step further, by suggesting the shift of control to the extensions of specific operating system or even dedicated applications; this is exactly what the "application server" concept is all about.

This article attempts to go a bit further than the SAN - DAS comparison. We use DAS as an alternative to SAN in order to show that life exists without SAN, as is a documented case in the small business and even large portion of middle size business marketplace It really questions the separation of STORAGE from the rest of computing infrastructure.

This paper is intended to inspire out of the box thinking from its readers. There is much controversy and differences of opinion on NAS versus DAS versus SAN, which may never be resolved. It provides much needed clarification on the need for metadata about the data which is really what sharing and RESTORE are all about.

**About authors:**

**Richard Blaschke** has over 40 years in the high tech industry including over 20 years in senior and executive management. He was seventeen years with IBM in both technical and managerial positions. He held senior managerial positions with several startup companies including Fortune Systems, the first company to introduce UNIX to the end user public, in 1981.

Dick spent the last 20 years specializing in Storage Products with companies like Amdahl as their VP and General Manager of their Peripheral Products Division, EMC as their world wide VP of Marketing, and Xiotech as their Senior VP of Marketing and Sales.

Currently Dick is on the board of Mitem Corp. in Manlo Park Ca., and the technical advisory board for Sherwood Information Partners in Denver Co.

**Samek Mokryn** is a founder and president of HalStor Inc. (www.halstor.com), design and consulting company specializing in storage and communication interfaces and architectures. He founded HalStor Inc. after more than 30 years of experience in product development, design and project management.

Samek also founded SANgate Systems - a storage appliance company - where he invented SANgate Systems' breakthrough technologies for managing stored data.
He founded C-Star Corp., a company that focused on development of new data storage technologies. C-Star later became SANgate Systems.

Samek previously held many senior-level technical positions with EMC Corp., including leading the development team for ESCON connectivity within EMC's Symmetrix Information Storage Systems. He also had an influence on EMC’s product and marketing policies. Before joining EMC, he served in various development roles related to I/O controllers and computer systems.

Samek holds an MSEE degree from Columbia University and earned a BSEE degree from Technion of Haifa, Israel.
He holds two patents in the area of a data replication.